**# assignment 4**

**#import the package numpy**

import numpy as np

**#import the package numpy**

import pandas as pd

**# read the ionosphere dataset using method read\_csv**

data = pd.read\_csv('ionosphere.data',header=None)

**# print the dimension of the data set**

print(data.shape)

**#removing the last colomn of the data matrix using the python code data\_new = data.iloc[:,:-1]**

data\_new = data.iloc[:,:-1]

**# after printing the dimensions of the dataset i got the shape of dimension is 351**

rows = 351

total = rows \* rows

**# import distance from scipy.spatial**

from scipy.spatial import distance

**# we are creating new matrix after removing last colomn of the data set**

data= np.matrix(data\_new)

**# transpose the data**

data=data.transpose()

**#compute the manhattan distance**

data\_man = np.arange(total,dtype=np.float).reshape(rows,rows)

for x in range(34):

for y in range(34):

data\_man[x,y] = (distance.cityblock(data[x], data[y]))

print(data\_man)

**#compute the euclidean distance**

data\_euc = np.arange(total,dtype=np.float).reshape(rows,rows)

for x in range(34):

for y in range(34):

data\_euc[x,y] = (distance.euclidean(data[x], data[y]))

print(data\_euc)

**#compute the minkowski distance**

data\_min = np.arange(total,dtype=np.float).reshape(rows,rows)

for x in range(34):

for y in range(34):

data\_min[x,y] = (distance.minkowski(data[x], data[y], 3))

print(data\_min)

**#compute the chebyshev distance**

data\_che = np.arange(total,dtype=np.float).reshape(rows,rows)

for x in range(34):

for y in range(34):

data\_che[x,y] = (distance.chebyshev(data[x], data[y]))

print(data\_che)

**#compute the cosine distance**

data\_cos = np.arange(total,dtype=np.float).reshape(rows,rows)

for x in range(34):

for y in range(34):

data\_cos[x,y] = (distance.cosine(data[x], data[y]))

print(data\_cos)

**#compute the jaccard distance**

data\_jac = np.arange(total,dtype=np.float).reshape(rows,rows)

for x in range(34):

for y in range(34):

data\_jac[x,y] = (distance.jaccard(data[x], data[y]))

print(data\_jac)



